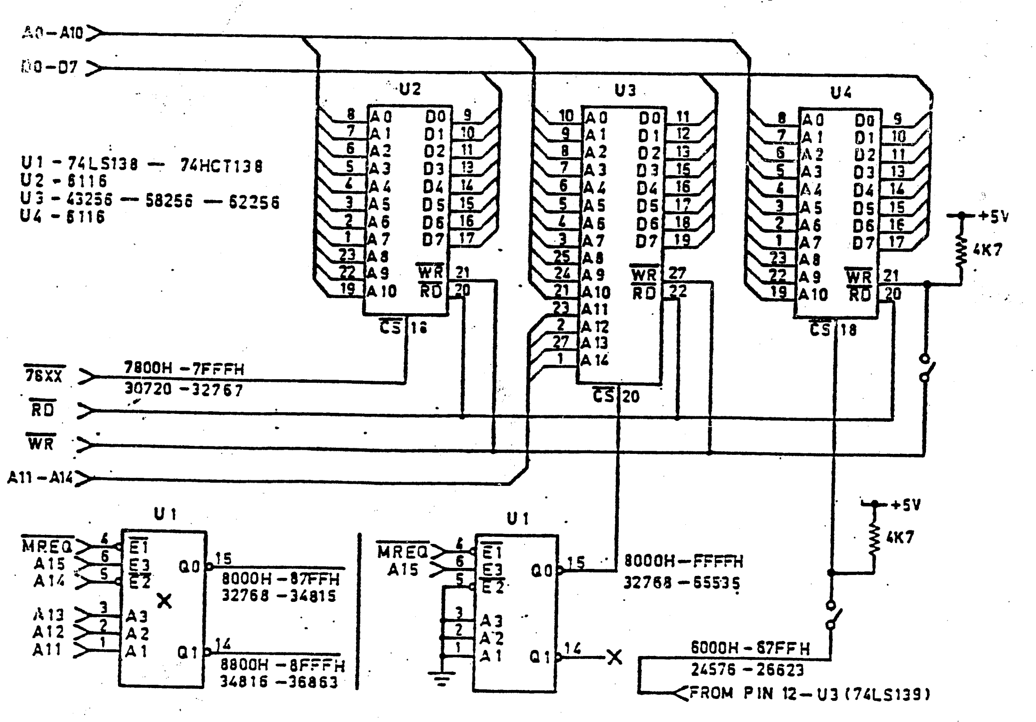
SEPT/OCT 1989 ISSUE # 26

VZ 200/300

HUNTER VALLEY VZ JOURNAL



PRODUCED BI-MONTHLY BY H.V.VZ.U.G.

THIS PUBLICATION WAS PREPARED ON A STAR NX 1000 PRINTER USING DAVE MITCHELL'S PATCH 3.3 WITH E & F WORD PROCESSOR. HI & LO-RES SCREEN DUMPS AND LISTINGS WERE DONE USING LARRY TAYLOR'S PRINTER PATCH V1.4 AVAILABLE FROM VSOFTWAREZ WHILE PATCH3.3 IS AVAILABLE FROM HUNTER VALLEY VZ USERS' GROUP.

FRONT COVER - IT SHOULD TELL YOU THAT I'VE BEEN USING A SOLDERING IRON AGAIN AFTER A FORCED SPELL OF MANY MONTHS AND IT'S A PLEASURE I'VE MISSED.

NOTE - THIS SUPPLEMENT LIKE THE PREVIOUS ONE IS DESIGNED TO BE REMOVED FROM JOURNAL AND PLACED WITH PREVIOUS SUPPLEMENT AND IS NUMBERED ACCORDINGLY.

VZ JOURNAL SUPPLEMENT -----------

FOR THIS AND THE NEXT COUPLE ISSUES AT LEAST THE SUPPLEMENT WILL CONTINUE. PART II OF IPL SEQUENCE DECODING IN NEXT ISSUE THEN IT'S AN EXPANDED COMMUNICATION REGION WITH DOS SYSTEM INCLUDED. THIS ISSUE CONTAINS A WHOPPING 23 PAGES, LOTS MORE THAN ANY OTHER VZ PUBLICATION.

44 WAY R. A. EDGE CONNECTORS ---

THERE'S MILLIONS OF COMMODORE 64'S AND TENS OF THOUSANDS OF VZ'S AROUND YET UP TO NOW 44 WAY R.A. (RIGHT ANGLE) EDGE CONNECTORS WERE VERY DIFFICULT TO OBTAIN. AT LONG LAST D. SMITH STOCKS 44 WAY R.A. EDGE CONNECTORS LIKE THE ONES IN YOUR MEMORY EXPANSION CARTRIDGE. I PAID $2.95 AT NEWCASTLE TIGHES HILL STORE. PRICES AND AVAILABILITY COULD VARY FROM STORE TO STORE SO CHECK. 30 WAY R.A. E/C'S ?, SORRY YOU STILL HAVE TO ORDER THEM IN FROM D.SMITH'S

JESMOND N'HOOD CENTRE OPEN DAY -

JESMOND NEIGHBOURHOOD CENTRE WILL HAVE AN OPEN DAY ON 28-10-89 (SATURDAY) APPROXIMATELY 10AM TO 3.30PM. THE IDEA IS FOR USERS OF ABOVE HALL TO PUT ON A SHOW OR DEMONSTRATION.

BESIDES H.V.VZ.U.G., KARATE CLUB, AEROBICS GROUP THERE WILL BE OTHER GROUPS AS WELL DOING THEIR BEST TO ENTERTAIN VISITORS AND ATTRACT NEW MEMBERS. YOUR SUPPORT IS NEEDED. BRING YOUR FAMILY, FRIENDS, ETC. IF THEY'RE NOT INERESTED THEY COULD DO SOME CHRISTMAS SHOPPING AT STOCKLAND MALL JESMOND.

NOTE - IF YOU CAN HELP MAN OUR STAND OR OTHERWISE THEN LET JOE LEON KNOW. THE OPEN DAY WILL BE FUND RAISER FOR THE CENTRE.

DISK DRIVE POWER SUPPLY HINTS --

NOT MANY PERSONS REALISE IT BUT TWO DRIVES CAN BE POWERED FROM THE ONE POWER SUPPLY. THE SECOND DRIVE GETS ITS POWER SUPPLY FROM BACK OF DISK CONTROLLER VIA THE RIBBON CABLE. I'VE BEEN USING THE ONE POWER SUPPLY WITH TWO DRIVES FOR OVER A YEAR NOW WITH NO PROBLEMS.

IT SEEMS VZ DISK DRIVE POWER SUPPLIES ARE EXTREMELY HARD TO GET. IF ENOUGH INTEREST IS SHOWN BY READERS I WOULD PREPARE A PROJECT FOR THE JOURNAL. THE POWER SUPPLY NEEDS +12 VDC AND +5 VDC AND ISN'NT A DIFFICULT PROJECT FOR ANYONE TO CONSTRUCT. PLEASE LET ME KNOW YOUR INTEREST.

SPEECH SYNTHESISER - OCT. MEETING

MY THANKS TO NEVILLE HUGHES FOR LOANING HIS UNIT TO H.V.VZ.U.G. FOR OUR PLANNED DEMONSTRATION AND AS IT WORKS OUT IT WILL ALSO BE HANDY FOR OPEN DAY AT THE CENTRE. I WOUL'NT BE SURPRISED IF OUR NEXT PRESIDENT WAS A VZ RUNNING THE SHOW VIA ITS SPEECH SYNTHESISER AND IT WOULD'NT BE SUCH A BAD IDEA AS ONLY A VZ WOULD KNOW EVEYTHING ABOUT A VZ. I HATE KNOW ALLS !!!

USING ROBERT QUINN'S PUT/GET*6 . . 26/4

IT SEEMS SOME OF OUR READERS, MYSELF INCLUDED HAD PROBLEMS INCORPORATING OR ADAPTING ROBERT'S PUT/GET*6 ROUTINE FOR USE WITH THEIR OWN PROGRAMS WHICH APPEARED IN ISSUE # 24, PAGES 11-12 SO I SPENT SOME TIME LEARNING HOW TO USE IT. THE RESULTING PROGRAM WILL APPEAR IN A FUTURE ISSUE.

YOU DON'T HAVE TO UNDERSTAND HOW THE M/C ROUTINE WORKS TO MAKE USE OF IT. ALL WE HAVE TO LEARN IS HOW TO USE PUT & GET AND EASIEST WAY IS TO EXPLAIN PARTS OF M/C ROUTINE WHICH FOLLOWS :-

```
POKE 31348, 33
POKE 31349,  0 - LO-BYTE - SOURCE ADDRESS (MOVE FROM)
POKE 31350,112 - HI-BYTE - 7000H - 28672
POKE 31351, 17 -
POKE 31352,  0 - LO-BYTE - DESTINATION ADDRESS (MOVE TO)
POKE 31353,192 - HI-BYTE - C000H - 49152
POKE 31354,  1
POKE 31355,  0 - LO-BYTE - NUMBER OF BYTES TO MOVE
POKE 31356,  8 - HI-BYTE - 800H - 2048
POKE 31357,237
POKE 31358,176
POKE 31359,201 - RETURN TO PROGRAM

POKE 30862,116 - LO-BYTE - START OF M/C PUT/GET*6 ROUTINE
POKE 30863,122 - HI-BYTE - 7A74H - 31348
```

THERE IS NOTHING MYSTERIOUS WITH ABOVE ROUTINE, IT'S JUST A SIMPLE MEMORY BLOCK MOVE ROUTINE AND KNOWING WHERE TO POKE SOURCE, DESTINATION AND # OF BYTES TO MOVE MAKES IT EASY TO USE. IF WE LOOK AT ABOVE WE'LL LEARN THAT IT'S SET TO MOVE 7000H (VIDEO RAM) TO C000H (49152) HI-MEMORY AND IT'S JUST AS EASY TO TO BRING IT BACK FROM C000H TO VIDEO RAM. IE :-

```
POKE31350,112:POKE31353,192 - MOVE VIDEO RAM TO C000H
POKE30862,116:POKE30863,122:POKEUSR(3),0

POKE31350,192:POKE31353,112 - MOVE C000H TO VIDEO RAM
POKE30862,116:POKE30863,122:POKEUSR(3),0
```

AS THE NUMBER OF BYTES BEING MOVED REMAINS CONSTANT THERE IS NO NEED TO POKE CHANGE IT. PLEASE ALSO NOTE THAT ONLY SOURCE AND DESTINATION HI-BYTES ARE BEING USED BECAUSE IN BOTH INSTANCES THE LO-BYTES ARE '0' AND REMAIN CONSTANT AS DO FOR STORES 1-8. SEE 64K RAM MEMORY MAP FOR EXAMPLE IN WHAT I MEAN.

THE PROGRAM MENTIONED EARLIER WILL INCORPORATE ROBERT QUINN'S PUTGET*8 MODIFIED FOR USE WITH 64K RAM PACK WHICH WILL ALLOW STORING OF 24 X 2K HI-RES AS PER 64K HI-RES SCREEN MEMORY MAP.

FOR SALE OR SWAP :-

ONE OFF VZ DISK DRIVE AND DISK CONTROLLER FOR SALE OR SWAP FOR A PRINTER TO GO WITH VZ 300 ALSO SOME PRINTING PAPER. FOR MORE INFORMATION CONTACT :-

RAY THOMPSON
SITE 220
MULLOWAY ROAD
CHAIN VALLEY BAY
NSW 2259

RELOCATING DOS C/R BY JOE LEON . .

AND DAVE MITCHELL

```
10 GOSUB 65000
20 CLS:COLOR,0:PRINT@35,"NEW TOP OF MEMORY =";
30 PRINT PEEK(30897)+256*PEEK(30898)
40 END
90 :
65000 POKE31148,201:IFPEEK(30897)+256*PEEK(30898)=48840'RETURN
65010 POKE 31148,195:POKE 31149,245:POKE 31150,54:COLOR,1
65020 CLS:PRINT@484,"OPEN DOOR & PRESS RETURN";
65030 IF INKEY$<>CHR$(13) THEN 65030
65040 CLS:POKE30897,255:POKE30898,191
65050 POKE30862,4:POKE30863,64:POKEUSR(3),0:RETURN
```

LOWERING TOM (TOP OF MEMORY) IS EASY ENOUGH TO ACHIEVE, BUT NOT MANY BASIC PROGRAMMERS KNOW HOW TO RELOCATE DOS C/R (COMMUNICATION REGION) BELOW IT WHICH IS ALSO FAIRLY EASY. BASICALLY TOM IS LOWERED FIRST AND THEN DOS IS INITIALISED BY A CALL TO 4004H (16388) BY A USR FUNCTION.

AFTER DOS IS REINITIALISED THE PROGRAM BREAKS TO BASIC WITH DOS AND READY MESSAGES BEING PRINTED OUT AS ON POWER UP. IT WOULD BE LOT NICER IF PROGRAM COULD CONTINUE AND THE ROUTINE ABOVE DOES JUST THAT. THE ROUTINE IS COMPLETELY RELOCATEABLE AND CONSISTS OF LINES 65000-65050 AND MUST BE CALLED FIRST BEFORE ANY OTHER OR ANY VARIABLES HAVE BEEN DECLARED AND IS VARIABLE FREE AND CAN BE USED AS PART OF ANY BASIC PROGRAM.

PROGRAM DISASSEMBLY :-

LINE 65000 - POKE 31148,201 DISABLES AUTORUN FUNCTION. WHEN PROGRAM IS RUN FOR FIRST TIME TOM CHECK IN LINE 65000 WILL NOT EQUAL 48840 SO IT WILL PROCEED TO NEXT LINE.

LINE 65010 - THIS LINE PLAYS A CRUCIAL PART BY SETTING UP AN AUTO RUN FUNCTION AND WAS SUPPLIED BY DAVE MITCHELL.

LINE 65020 - THIS LINE SIMPLY GIVES THE USER OPPORTUNITY TO OPEN DRIVE DOOR THUS AVOIDING POSSIBLE CORRUPTION OF DISK.

LINE 65030 - AWAITS USER TO PRESS RETURN. IF BREAK IS USED THEN PROGRAM WILL RERUN AND IS CAUSED BY LINE 65010.

LINE 65040 - TOM IS LOWERED TO 49151 BY THIS LINE.

LINE 65050 - THIS LINE INITIALISES THE DOS AND RELOCTES IT BELOW 49151 WHICH WAS THE TOM SET BY PREVIUS LINE.

NOTE - ONCE DOS IS RESET PROGRAM AUTORUNS AND GOSUBS TO LINE 65000 FOR THE SECOND TIME. BECAUSE TOM CHECK NOW EQUALS 48840 PROGRAM NOW RETURNS TO LINE 20 AND CONTINUES WITH PROGRAM. IT DOES'NT MATTER HOW MANY TIMES YOU BREAK AND RUN AGAIN, PROGRAM WILL CONTINUE AS NORMAL UNLESS YOU ALTER TOM OR RESET THE VZ.

LINE 20 & 30 - PRINTS NEW TOM WHICH NOW IS 48840.

TO WORK OUT THE TOM CHECK NUMBER IN LINE 65000 IS SIMPLE. FOR EXAMPLE, YOU WANT TO SET NEW TOM TO 47103. JUST SUBTRACT 311 BYTES AND YOU'LL GET YOUR TOM CHECK NUMBER WHICH WILL BE 46792.

BANK 1 - STORE 8 F800H - FFFFH 63488 - 65535	BANK 2 - STORE 8 F800H - FFFFH 63488 - 65535	BANK 3 - STORE 8 F800H - FFFFH 63488 - 65535
BANK 1 - STORE 7 F000H - F7FFH 61440 - 63487	BANK 2 - STORE 7 F000H - F7FFH 61440 - 63487	BANK 3 - STORE 7 F000H - F7FFH 61440 - 63487
BANK 1 - STORE 6 E800H - EFFFH 59392 - 61439	BANK 2 - STORE 6 E800H - EFFFH 59392 - 61439	BANK 3 - STORE 6 E800H - EFFFH 59392 - 61439
BANK 1 - STORE 5 E000H - E7FFH 57344 - 59391	BANK 2 - STORE 5 E000H - E7FFH 57344 - 59391	BANK 3 - STORE 5 E000H - E7FFH 57344 - 59391
BANK 1 - STORE 4 D800H - DFFFH 55296 - 57343	BANK 2 - STORE 4 D800H - DFFFH 55296 - 57343	BANK 3 - STORE 4 D800H - DFFFH 55296 - 57343
BANK 1 - STORE 3 D000H - D7FFH 53248 - 55295	BANK 2 - STORE 3 D000H - D7FFH 53248 - 55295	BANK 3 - STORE 3 D000H - D7FFH 53248 - 55295
BANK 1 - STORE 2 C800H - CFFFH 51200 - 53247	BANK 2 - STORE 2 C800H - CFFFH 51200 - 53247	BANK 3 - STORE 2 C800H - CFFFH 51200 - 53247
BANK 1 - STORE 1 C000H - C7FFH 49152 - 51199	BANK 2 - STORE 1 C000H - C7FFH 49152 - 51199	BANK 3 - STORE 1 C000H - C7FFH 49152 - 51199
BANK 0 - BUFFER B800H - BFFFH 47104 - 49151		
DOS COM. REGION B6C9H - B7FFH 46793 - 47103		
TOP OF MEMORY B6C8H - 46792 USER MEMORY		

USER MEMORY BANK 0 R A M 8000H - BFFFH 32768 - 49151
RAM COM. REGION 7800H - 7AE8H 30720 - 31464
2K VIDEO RAM 7000H - 77FFH 28672 - 30719

	HEX	DEC	LO	HI BYTE
STORE 8	F800	63488	000	248
STORE 7	F000	61440	000	240
STORE 6	E800	59392	000	232
STORE 5	E000	57344	000	224
STORE 4	D800	55296	000	216
STORE 3	D000	53248	000	208
STORE 2	C800	51200	000	200
STORE 1	C000	49152	000	192
BUFFER	B800	47104	000	184
VIDEO RAM	7000	28672	000	112
# BYTES	0800	2048	000	008

BANK 0 RAM	8000H - BFFFH	32768 - 49151
BANK 1 RAM	C000H - FFFFH	49152 - 65535
BANK 2 RAM	C000H - FFFFH	49152 - 65535
BANK 3 RAM	C000H - FFFFH	49152 - 65535

64K R/PACK & HI-RES SCREENS CONT. 26/7

TALKING ABOUT BANK SWITCHING, VIDEO STORES 1, 2, 3, ETC. CAN BE CONFUSING SO I'VE PREPARED A 64K RAM PACK MEMORY MAP SO IT WOULD BE EASIER FOR READERS TO UNDERSTAND.

ONE OF THE MORE OBVIOUS USES OF 64K OR 128K S/WAYS RAM IS TO STORE 2K HI-RES OR 6K SUPER HI-RES SCREENS. USING 64K R/PACK AS EXAMPLE, 24 X 2K OR 8 X 6K HI-RES SCREENS CAN BE STORED IN MEMORY FOR USE IN SLIDE SHOW FOR ADVERTISING, ETC.

ONLY BANKS 1 TO 3 CAN BE SWITCHED WITH BANK 'Ø' BEING FIXED. IT OCCUPIES MEMORY LOCATIONS 8ØØØH-BFFFH (32768-49151). BANKS 1 TO 3 ALL SHARE THE SAME MEMORY LOCATION - CØØØH-FFFFH (49152-65535), BUT ONLY ONE AT A TIME. BOTH SOFTWARE AND HARDWARE IS USED IN SWITCHING BANKS.

I/O PORT 112-127 IS RESERVED FOR BANK SWITCHING AND ANY NUMBER IN ABOVE RANGE CAN BE USED TO SWITCH BANKS, IE :-

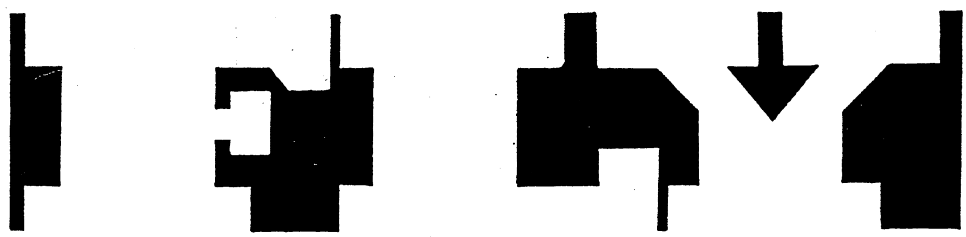
OUT 112,2 OR OUT 113,2 OR . . . OUT 126,2 OR OUT 127,2

BOTH TAPE AND D/DRIVE USERS CANNOT SWITCH BANKS WILLY-NILLY AS THEY SHOULD BE AWARE OF THE FOLLOWING. ON POWER UP OR RESET BANK '1' IS THE DEFAULT BANK AND TOM TOGETHER WITH ALL MEMORY LOCATIONS FOR HOUSEKEEPING IS SET AND IN CASE OF D/DRIVE USERS DOS C/R (COMMUNICATION REGION) IS SET AT TOP OF BANK '1'.

IF WE USED THE OUT COMMAND TO SWITCH BANKS THEN A SYSTEM CRASH WOULD OCCUR BECAUSE WE WOULD BE DESTROYING THE LINKS BETWEEN DOS AND RAM C/R. THIS SITUATION CAN BE AVOIDED BY LOWERING TOM BELOW BANK '1' AND IN CASE OF DOS RELOCATING ITS C/R BELOW IT. SEE ARTICLE ON PAGE 5 ON HOW TO ACCOMPLISH IT.

THE MEMORY MAP, R. QUINN'S P/GET AND RELOCATING DOS C/R SHOULD GIVE READERS AN INSIGHT ON HOW TO USE THE IDEAS IN THEIR OWN PROGRAMS. HAVE FUN !!!

GRAPHIC PUZZLE - CAN YOU SOLVE IT ? ? ?



BOOLEAN LOGIC FUNCTIONS AND HI-RES GRAPHICS BY BOB KITCH

LOGIC OPERATIONS, IN EITHER ASSEMBLER OR BASIC, ARE POORLY UNDERSTOOD BY PROGRAMMERS. THEY ARE HOWEVER MORE FUNDAMENTAL THAN ARITHMETIC OPERATORS (+ - * / ^) AND RELATIONAL OPERATORS (< > = <>).

THE ONLY DIFFERENCE BETWEEN LOGIC OPERATORS IN BASIC AND ASSEMBER IS THE LENGTH OF THE BIT STREAMS (BYTES OR WORDS) UPON WHICH EACH OPERATES. IN ASSEMBLER, ALL LOGIC FUNCTIONS INVOLVE 8-BIT OPERANDS (SINGLE BYTE), ONE OF WHICH MUST BE STORED IN THE A-REGISTER. THE RESULT IS ALWAYS STORED BACK IN THE A-REGISTER. ALL BASIC LOGIC FUNCTIONS OPERATE ON 16-BIT INTEGER VALUES.

THE Z80 INSTRUCTION SET SUPPORTS FOUR LOGIC OPERATORS - AND, OR, XOR AND CPL. IN BASIC THREE LOGIC OPERATORS ARE AVAILABLE - AND, OR AND NOT. (COMPLEMENT AND NOT ARE THE SAME OPERATION).

LOGIC OPERATORS ONLY MAKE "SENSE" WHEN THE OPERANDS ARE CONVERTED TO BINARY NOTATION. THIS IS THE SINGLE LARGEST PROBLEM THAT PROGRAMMERS HAVE IN GETTING TO GRIPS WITH LOGIC OPERATORS. FURTHERMORE, LOGIC OPERATORS WORK ON A SINGLE BIT AT A TIME. UNLIKE ARITHMETIC FUNCTIONS, IN WHICH CARRYING OR BORROWING EFFECT ADJACENT BITS, LOGIC FUNCTIONS ARE "BLIND" TO BITS IN COLUMNS NEXT TO THE BIT BEING TESTED AND CHANGED. WITH THE EXCEPTION OF NOT (AND CPL) EACH LOGIC FUNCTION REQUIRES TWO OPERAND BITS TO PRODUCE A SINGLE BIT RESULT. THE ORDER OF OPERANDS DOES NOT EFFECT THE RESULT EITHER. (COMPARE THIS TO ARITHMETIC OR RELATIONAL OPERATORS).

A TRUTH TABLE FOR THE FOUR LOGIC FUNCTIONS IS AS FOLLOWS -

X	Y	X AND Y	X OR Y	X XOR Y	NOT X
0	0	0	0	0	1
0	1	0	1	1	1
1	0	0	1	1	0
1	1	1	1	0	0

TO CONVERT DECIMAL TO BINARY NOTATION SEE MY ARTICLE IN AEM DEC. 1986 PG. 92. MY PROGRAM IN HVVZUG NEWSLETTER 1(12) MAY/JUN 1987 PG. 7-9 IS A USEFUL TRAINER AND INTRODUCTION TO LOGIC OPERATORS. I COMMEND BOTH ARTICLES TO YOU FOR YOUR INTEREST.

TO STRESS THE IMPORTANCE OF CONVERTING NUMBERS TO BINARY WHEN TRYING TO COMPREHEND LOGIC OPERATORS, CONSIDER THE FOLLOWING -

9 AND 12 = ?? (DOESN'T MEAN MUCH??)

BUT...

```
00001001B      9
00001100B AND 12
--------- ------
00001000B      8  IS A LITTLE CLEARER!
```

LOGIC IN ASSEMBLY LANGUAGE

THIS MAY BE THOUGHT OF AS "WORKING WITH BITS". THE Z80 ALLOWS US TO TURN BITS ON (SET OR 1) AND OFF (RESET OR 0), OR TEST (BIT) FOR THE TWO CONDITIONS. BITS CAN BE SHIFTED OR "SLID", OR ROTATED AROUND IN A LOOP. LASTLY THE PROGRAMMER CAN ADD OR REMOVE BITS DEPENDING ON THE RELATIONSHIP WITH OTHER BITS IN THE BYTE. PRETTY POWERFUL SET OF BIT MANIPULATING INSTRUCTIONS.

AND IS MOST OFTEN USED AS A BIT MASK, DELETING (RESETTING) UNWANTED BITS AND PRESERVING THOSE IN A SPECIFIED PART OF A BYTE. THE GOLDEN RULE FOR THIS OPERATION IS -

"WHEREVER A ZERO OCCURS IN THE MASK, THE CORRESPONDING BIT WILL ALSO BE ZERO. WHEREVER A ONE OCCURS IN THE MASK, THE CORRESPONDING BIT WILL REMAIN UNCHANGED."

SUPPOSE THAT WE ARE CONCERNED WITH THE LAST FOUR BITS (BITS 0 TO 3) OF A BYTE (I.E. MASK OFF THE MOST SIGNIFICANT FOUR BITS). IT IS ONLY NECESSARY TO AND THE BYTE WITH 0FH (00001111B) TO ERASE ANY BITS IN THE FIRST FOUR POSITIONS (BITS 4 TO 7). IN ASSEMBLY LANGUAGE THIS OPERATION LOOKS LIKE THIS -

```
LD A,BYTE
AND 0FH
```

THE ONLY OTHER ALTERNATIVE, IS TO PERFORM FOUR SEPARATE RES OPERATIONS, TO ELIMINATE THE FIRST FOUR BITS - NOT VERY EFFICIENT.

OR, ON THE OTHER HAND, IS MOST OFTEN USED TO TO SET A GROUP OF BITS AT ONCE. THE GOLDEN RULE FOR THIS OPERATION IS -

"WHEREVER A ONE APPEARS IN THE OPERANDS, A ONE WILL APPEAR IN THE RESULT."

IF IT IS REQUIRED TO ENSURE THAT BITS 0 TO 3 ARE SET THEN YOU COULD OR 0FH INSTEAD OF PERFORMING FOUR SET OPERATIONS -

```
LD A,BYTE
OR 0FH
```

ANOTHER COMMON USE OF THE OR OPERATOR IS TO CHECK IF A 16-BIT REGISTER PAIR IS ZERO -

```
LD A,B
OR C
```

THE A-REGISTER WILL ONLY HOLD ZERO (AND THE ZERO FLAG SET) IF REGISTERS B AND C ARE BOTH ZERO. (CHECK IT TO CONVINCE YOURSELF!).

THE AND AND OR OPERATORS CAN BE USED TOGETHER TO FORCE A VALUE INTO AN APPROPRIATE RANGE. FOR EXAMPLE, GRAPHIC ROUTINES OFTEN NEED TO ENSURE THAT THE CURRENT VALUE IN THE HL-REGISTER IS A LEGITIMATE SCREEN ADDRESS. SINCE THE SCREEN IS MEMORY MAPPED FROM 7000H TO 77FFH, IF THE VALUE IN THE H-REGISTER IS BETWEEN 70H AND 77H, THEN THE HL-REGISTER MUST BE POINTING TO THE SCREEN. TO FORCE THE H-REGISTER INTO THAT RANGE, WHILE MAINTAINING ITS CURRENT VALUE IF IT IS ALREADY IN THE CURRENT RANGE, THE FOLLOWING COULD BE USED -

```
LD A,H      ;PUT MSB INTO A-REG.
AND 77H     ;ELIMINATE BITS 3 AND 7.
OR 70H      ;SET BITS 4, 5 AND 6.
LD H,A      ;H-REG IS IN CORRECT RANGE FOR VRAM.
```

XOR IS MOST OFTEN USED IN ASSEMBLY LANGUAGE TO CLEAR THE A-REGISTER, SINCE ANY VALUE XORED WITH ITSELF IS ZERO. (CHECK IT!) WHENEVER XOR A IS SEEN IN A LISTING, THE PROGRAM IS CLEARING THE A-REGISTER TO ZERO, THE ZERO FLAG IS SET AND THE CARRY FLAG IS RESET. THAT'S A LOT OF WORK FOR A SINGLE FAST INSTRUCTION TO PERFORM. SUB A OR LD A,0 ALL CLEAR THE A-REGISTER. LD A,0 TAKES MORE TIME AND MEMORY BUT DOES NOT AFFECT THE STATUS FLAGS.

THE XOR OPERATION IS QUITE A FUSSY INSTRUCTION HOWEVER, AND HAS OTHER USES. IT CAN MERGE BYTES TOGETHER BUT WILL EXCLUDE PAIRS OF BITS. IT CAN BE USED TO ELIMINATE SIMILARITIES. IT IS OFTEN USED AS A "TOGGLE SWITCH" SUCH AS WHEN A ROUTINE IS ONLY TO BE EXECUTED EVERY SECOND TIME IT WAS CALLED. (USEFUL FOR FLASHING DISPLAYS).

A COMMON USE FOR THE TOGGLING ACTION IS TO SWITCH THE OUTPUT LATCH ON THE VZ TO ACTUATE THE SPEAKER. THE WRITE-ONLY LATCH IS MAPPED INTO 6800H - 6FFFH. BITS 5 AND 0 ARE THE DRIVERS FOR THE SPEAKER AND MUST BE OF COMPLIMENTARY VALUE TO PUSH-PULL THE SPEAKER. OTHER BITS IN THE LATCH CONTROL OTHER FUNCTIONS AND SHOULD NOT BE CHANGED WHEN THE SPEAKER IS DRIVEN. THE MASK THAT ACHIEVES THIS ACTION IS 00100001B OR 21H. IF THE LATCH CONTAINS A VALUE OF 00110100B (34H) THEN TOGGLING (XORING) IT WITH 21H (THE MASK VALUE) WILL RESULT IN THE LATCH CHANGING BETWEEN 34H (00110100B) AND 15H (00010101B). THIS RESULTS IN BITS 0 AND 5 BEING SWITCHED ON AND OFF AS REQUIRED AND THE OTHER BITS REMAIN UNCHANGED.

NOT OR CPL ONLY REQUIRE ONE OPERAND. IN THESE PROCESSES, EVERY BIT IS ALTERED AND EACH BIT IS "INVERTED" I.E. ZEROS BECOME ONES AND ONES BECOME ZEROS. THE MOST OFTEN APPLICATION OF THIS PROCESS IS IN "TWO COMPLEMENTS" ARITHMETIC. TO FORM A NEGATIVE INTEGER VALUE FROM A POSITIVE INTEGER, IT IS FIRST COMPLEMENTED AND THEN INCREMENTED BY ONE -

```
LD A,BYTE
CPL
INC A
```

PIXEL REPLACEMENT IN HI-RES SCREENS.

SOME TIME AGO, I HAD TO WRITE AN ASSEMBLER SUBROUTINE THAT ALLOWED A HI-RES SCREEN TO "SWEEP" OVER ANOTHER HI-RES SCREEN. THIS INVOLVED A "COLUMN-BY-COLUMN" REPLACEMENT OF THE "UNDERLYING" SCREEN ON A PIXEL-BY-PIXEL BASIS. AS HI-RES SCREENS ARE NOT BYTE-MAPPED (EACH PIXEL USES TWO BITS FOR REPRESENTATION) THIS TURNED OUT TO BE A NON-TRIVIAL EXERCISE. SOME ELEGANT USE OF LOGIC OPERATORS WAS CALLED INTO PLAY - NOT TO MENTION, SOME TIDY ALLOCATION OF THE Z80 REGISTER SET!

TO ILLUSTRATE THE NEAT MATHS OF THIS EXERCISE, CONSIDER THE FOLLOWING EXAMPLE. LET'S SUPPOSE THAT THE "INCOMING" SCREEN IS TO SWEEP FROM LEFT TO RIGHT OVER THE "REPLACED" SCREEN. FURTHERMORE, LET'S SAY THAT THE DISPLAYED BYTE ON THE SCREEN IS B1H (10 11 00 01B) AND THAT THE BYTE TO REPLACE THIS IS D8H (11 01 10 00B). (YOU CAN WORK OUT THE COLOURS OF THE FOUR PIXELS IN EACH INSTANCE.) THE SEQUENCE OF BYTES THAT NEED TO BE WRITTEN TO THE PARTICULAR SCREEN ADDRESS ARE AS FOLLOWS -

```
[10 11 00 01B  B1H ORIGINAL BYTE
 11[11 00 01B  F1H LEFT HAND PIXEL REPLACED
 11 01[00 01B  D1H NEXT PIXEL REPLACED
 11 01 10[01B  D9H NEXT
 11 01 10 00B[ D8H INCOMING PIXEL COMPLETE
```

(BITS TO THE RIGHT OF [ARE UNCHANGED OR NOT REPLACED.)

THE SEQUENCE IN HEX IS -
B1H - F1H - D1H - D9H - D8H.
HOW ON EARTH DO WE GET THIS SEQUENCE IN QUICK AND EFFICIENT ASSEMBLER CODE? (NOTICE HOW THE USE OF BINARY NOTATION IS EXTREMELY ILLUMINATING WHEN WORKING AT THIS LEVEL.)

USING SOME DEDUCTIVE MATHS THEORY, IT TURNS OUT THAT THE FOLLOWING LOGIC SEQUENCE PROVIDES THE DESIRED RESULT!

```
(B1H AND 3FH) OR (D8H AND NOT 3FH) = F1H
(F1H AND 0FH) OR (D8H AND NOT 0FH) = D1H
(D1H AND 03H) OR (D8H AND NOT 03H) = D9H
(D9H AND 00H) OR (D8H AND NOT 00H) = D8H
```

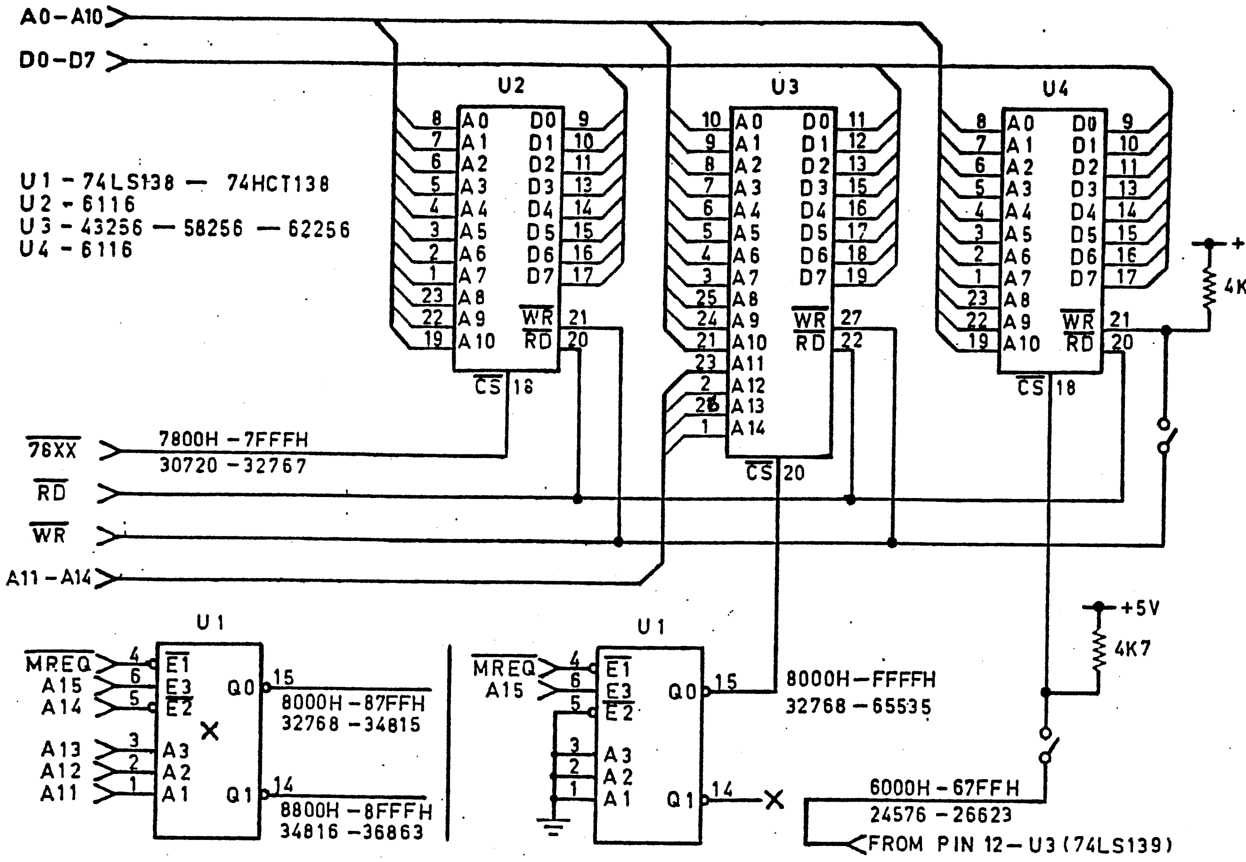
(SOME GOOD FORTRAN IV CODE THERE!)

THE LEFT-HAND BRACKET MASKS OUT THE LEFT-HAND PIXELS OF THE REPLACED BYTE, THE RIGHT-HAND BRACKET MASKS OUT THE RIGHT-HAND PIXELS OF THE INCOMING BYTE, AND THE TWO ARE LOGICALLY ADDED TOGETHER TO GIVE THE RESULT.

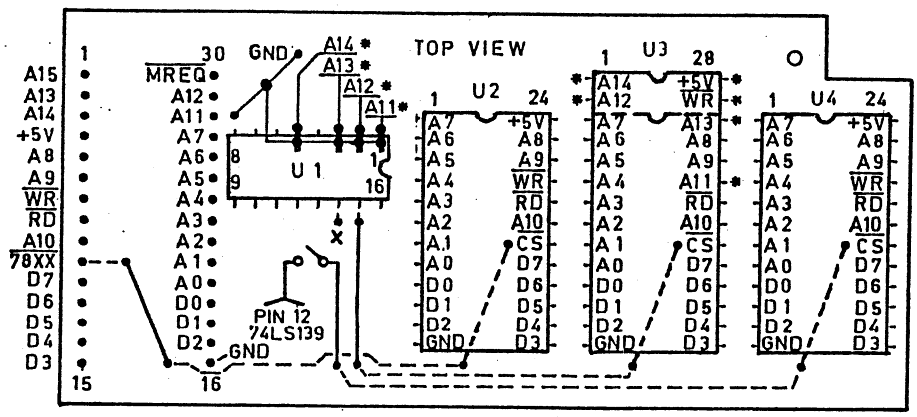
AS I SAID, THE MATHS IS BEAUTIFUL! THE INTERESTING THING IS THAT THE "INCOMING" BYTE (D8H) IS A CONSTANT AS IT IS HELD IN A MEMORY BUFFER SOMEWHERE, WHILST THE "REPLACED" BYTE VARIES AS IT IS WRITTEN TO THE SCREEN - WHICH IS AFTER ALL, ONLY AN AREA OF MEMORY. THE SEQUENCE OF MASKS IS NOW THE INTERESTING SERIES TO STUDY. THE MASK SERIES IS -
3FH - 0FH - 03H - 00H.

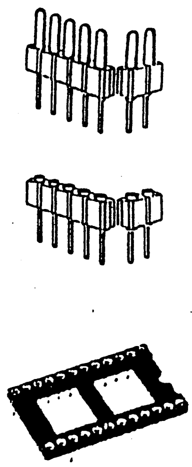
CONTINUED IN NEXT ISSUE . . .

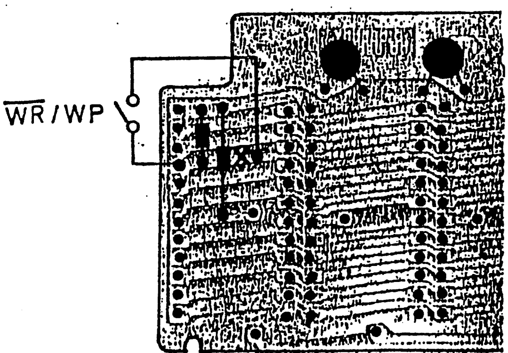
VZ 200 34K RAM MODS BY JOE LEON



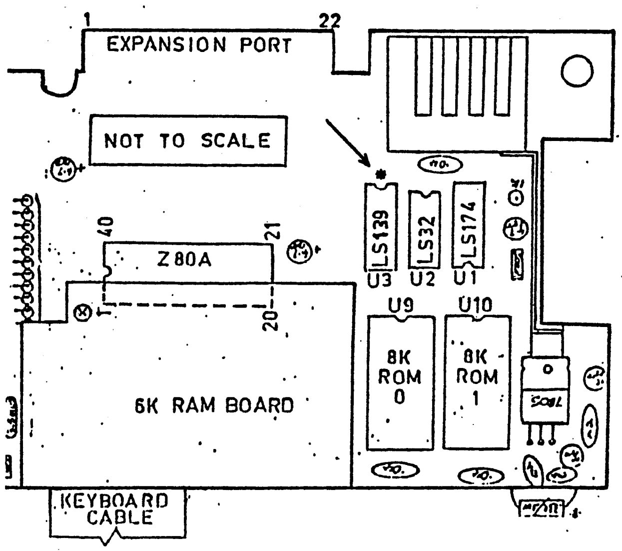
VZ 200 6K RAM BOARD







BOTTOM VIEW 6K RAM BOARD



THIS PROJECT IS DESIGNED TO PROVIDE VZ 200 OWNERS 34K OF RAM BY REPLACING ONE OF THE INBUILT 2K RAM CHIPS WITH A 32K RAM CHIP WHICH WILL TAKE MEMORY UP TO FFFFH (65535) AND AS A BONUS 2K RAM AT 6000H (2K GAP ABOVE DOS ROM). I PRESUME PERSONS ATTEMPTING MOD HAVE NECESSARY SKILLS AS THE RISKS ARE YOURS.

VZ 200'S ARE LOT EASIER TO WORK ON AND USE DISCRETE COMPONENTS AND CAN TAKE FAIR BIT OF ABUSE WHILE VZ 300'S USE A FEW DEDICATED CHIPS AND AS A RESULT MODS ARE HARDER AND I FOUND THEM VERY TOUCHY AND I'VE A FEW DEAD VZ 300'S TO ATTEST TO THAT. THE CIRCUIT AND DIAGRAMS TELL ABOUT ALL AND A BIT OF EXPLANATION WOUD'NT GO ASTRAY SO HERE GOES.

CIRCUIT DIAGRAM - THREE MODS WERE DONE TO ORIGINAL 6K CIRCUIT WHICH ARE U1 (74LS138), U3 (32K) & U4 (2K) RAMS. BY COMPARING U2-U4 YOU'LL NOTE U3 HAS FOUR EXTRA ADDRESS INPUTS AND IS A 28 PIN DEVICE.

U1 (74LS138) - THIS ADDRESS DECODER ORIGINALLY DECODED IN 2K BLOCKS AND BY GROUNDING PINS 1, 2, 3 & 5 IT BECOMES A 32K BLOCK DECODER. THE VACANT HOLES LEFT BY ABOVE GROUNDED PINS PROVIDE A11-A14 WHICH ARE CONNECTED TO 32K RAM CHIP, REAL HANDY. AS BEFORE PIN 15 SUPPLIES CS BAR SIGNAL TO U3.

U4 (6116) - THE 32K RAM MAKES U4 REDUNDANT BUT NOT USELESS AS IT CAN BE PRESSED INTO SERVICE FILLING THE 2K GAP AT 6000H. BY GOOD FORTUNE AN UNUSED DECODED OUTPUT FOR 6000H-67FFH RANGE IS AVAILABLE FROM U3 (74LS139), PIN 12 ON MAIN PCB, REAL HANDY THAT.

AT RIGHT SIDE OF PCB LAYOUT ARE SHOWN FROM TOP :-

1) DOUBLE SIDED MACHINED INSERT ADAPTOR STRIP
2) SINGLE IN LINE MACHINED INSERT SOCKET STRIP
3) 24 PIN DUAL IN LINE MACHINED INSERT SOCKET

BOTH 1 & 2 CAN BE SNAPPED INTO DESIRED LENGTHS AND I USE SINGLE/DOUBLE IN LINE M/I SOCKETS/STRIPS TOGETHER WITH ADAPTOR STRIPS TO MAKE PLUG/SOCKET ARRANGEMENTS. THIS APPROACH ALLOWS ME TO MAKE MOST OF MY PROJECTS REMOVEABLE FOR MODS OR FAULT FINDING.

CONSTRUCTION DETAILS - FIRST THE MAIN PCB WILL HAVE TO BE REMOVED FROM YOUR VZ 200 AND THEN ITS METAL RF SHIELD AS WELL. NEXT UNSOLDER OR CUT RIBBON CABLE SUPPORTING 6K RAM BOARD AND REMOVE REMAINS OF RIBBON CABLE FROM BOTH PCB'S.

IF YOU CAN GET A M/I SOCKET STRIP THEN BREAK IT IN TWO 15 PIN LENGHTS AND SOLDER TO MAIN PCB WHERE 6K RAM BOARD WAS MOUNTED VIA ITS RIBBON CABLE OR IF YOU CAN'T GET STRIP THEN CUT DOWN TO 30 PINS A 40 PIN M/I SOCKET AND SOLDER IT IN.

FIND U1 (74LS138) ON 6K RAM BOARD AND REMOVE LEFT LINK WIRE BELOW IT. CUT PINS 1, 2, 3 AND 5 ON U1 CLOSE TO PCB AND REMOVE STUBS FROM PCB AND THEN BEND CUT PINS UP AND OVER U1 AS SHOWN ON DIAGRAM. USING A SHORT LENGTH OF BARE WIRE SOLDER ALL BENT PINS TOGETHER AND OTHER END TO GROUND LINK WIRE JUST ABOVE U1.

UNSOLDER THE TWO CERAMIC CAPACITORS ABOVE U2 & U3 AND SOLDER TO REVERSE SIDE USING SAME HOLES. SOLDER TWO 4K7 RESISTORS TO BOTTOM OF PCB AS SHOWN ON BOTTOM LEFT DIAGRAM AND CUT TRACK AT SPOT SHOWN WITH AN X.

NEXT PROCEED TO REMOVE U3 (MIDDLE 2K RAM) EITHER BY UNSOLDERING IT OR CUTTING ITS PINS CLOSE TO PCB ON ONE SIDE AND LEVERING BACK AND FORTH TILL PINS BREAK ON OTHER SIDE. DON'T FORGET TO REMOVE PIN STUBS FROM BOARD AFTER. SOLDER A 24 PIN MACHINE INSERT SOCKET IN U3'S PLACE. SOLDER OR INSERT TWO SHORT LENGHTS OF WIRE IN 24 PIN SOCKET WHERE A11 & A13 ARE MARKED WITH AN ASTERICK.

USING SMALL PLIERS BEND UP PINS 1, 2, 23, 26, 27 & 28 ON 32K RAM CHIP JUST A LITTLE OVER 90 DEGREES BEING CAREFUL NOT TO TOUCH PINS AS STATIC FROM YOUR BODY COULD ZAP 32K RAM CHIP. INSERT 32K RAM CHIP NEXT. SOLDER WIRE FROM UNDER A13 TO PIN 28 AND WIRE FROM UNDER A11 TO PIN 27.

NOTE - 32K RAM COULD BE SOLDERED DIRECT TO BOARD IF DESIRED.

NEXT SOLDER FOUR LENGHTS OF WIRE TO HOLES BELOW U1 AT PINS 1, 2, 3 & 5 WHICH PROVIDE A11, A12, A13 & A14 AND SOLDER OTHER ENDS TO CORRESPONDING ADDRESS PINS ON 32K RAM.

USING TWO 15 PIN LENGHTS OF DOUBLE SIDED PIN ADAPTER STRIPS PUSH THE THIN PINS INTO 30 PIN SOCKET ON MAIN PCB AND LOWER 6K RAM BOARD OVER AND SOLDER PROTRUDING PINS. UNPLUG RAM BOARD AND CONNECT SWITCH TO REVERSE SIDE OF BOARD AS SHOWN ON BOTTOM LEFT DIAGRAM.

NEXT CONNECT ONE END OF WIRE FROM REMAINING SWITCH TO TOP OF BOARD BOTTOM HOLE WHERE THE LINK WIRE USED TO BE. LOCATE 74LS139 ON MAIN PCB USING BOTTOM RIGHT DIAGRAM AS A GUIDE AND SOLDER OTHER END OF WIRE TO PIN 12 OF 74LS139 OR A SINGLE BROKEN OF SOCKET COULD BE SOLDERED TO PIN 12 OF 74LS139 AND ANOTHER TO WIRE END MAKING BOARD COMPLETELY REMOVEABLE. AFTER TESTING MOUNT THE SWITCHES ANYWHERE CONVENIENT.

THAT COMPLETES THE MODS AND TESTING CAN BEGIN. CONNECT YOUR VZ 200 AS NORMAL, BUT WITHOUT DISK DRIVE AND POWER UP AND IF THE READY MESSAGE APPEARS THEN TYPE IN THE FOLLOWING :-

PRINT PEEK(30897)+256*PEEK(30898) AND PRESS RETURN

65535 SHOULD BE DISPLAYED INFORMING YOU NOW HAVE 34K USER RAM. ONE MORE TEST REMAINS AND THAT IS TO CHECK OUT 2K RAM AT 6000H (24576) AND BOTH SWITCHES MUST BE IN CLOSED POSITION FOR TEST. TYPE IN FOLLOWING AND PRESS RETURN.

POKE 24576,0:PRINT PEEK(24576)

IF '0' IS DISPLAYED THEN ALL OK AND VZ 200 CAN BE PUT BACK TOGETHER AGAIN AND NO LONGER WILL YOU HAVE TO PLUG RAM PACKS IN WITH ASSOCIATED PROBLEMS.

NOTE - WORDPRO CARTRIDGE WILL NOT WORK WITH THIS MOD AS IT REQUIRES NO RAM AT 6000H AND FROM D000H-FFFFH. INTERESTED PERSONS CAN CONTACT ME FOR MORE INFORMATION.

8K OR 32K RAM AT 6000H (4 X 2K OR 16 X 2K BANKS) :-
BY THE ADDITION OF TWO EXTRA CHIPS AND AN 8K OR 32K RAM CHIP THEY CAN BE EASILY INCORPORATED. PLEASE LET ME KNOW YOUR INTEREST AND I'LL PREPARE ARTICLE FOR BOTH ABOVE.

```
10 REM - SUITE 2 **************
15 :
20 REM - INITIALISER **********
25 A=(PEEK(30897)+PEEK(30898)*256)-450::B%=A/256:C%=A-B%*256
30 POKE 30897,C%:POKE30898,B%:CLEAR50
35 :
40 CLS:PRINT@485;"PLEASE WAIT 17 SECONDS";
45 :
50 A=PEEK(30897)+PEEK(30898)*256:E=65536:B%=31273
55 FORR=1TO73:READC%:B=B+C%:POKEB%,C%:B%=B%+1:NEXT
60 D=A+1-E:FORR=74TO521:READC%:B=B+C%:POKED,C%:D=D+1:NEXT
65 :
70 IF B<>45643THEN PRINT"ERROR IN DATA":SOUND30,2:END
95 :
100 D=A+21:B%=D/256:C%=D-B%*256:POKEA+43-E,C%:POKEA+44-E,B%
105 D=A+29:B%=D/256:C%=D-B%*256:POKEA+49-E,C%:POKEA+50-E,B%
110 D=A+268:B%=D/256:C%=D-B%*256:POKEA+202-E,C%:POKEA+203-E,B%
115 D=A+1:B%=D/256:C%=D-B%*256:POKE30846,C%:POKE30847,B%
120 D=A+317:B%=D/256:C%=D-B%*256:POKEA+228-E,C%:POKEA+229-E,B%
125 D=A+327:B%=D/256:C%=D-B%*256:POKE31125,C%:POKE31126,B%
130 D=A+420:B%=D/256:C%=D-B%*256:POKEA+371-E,C%:POKEA+372-E,B%
135 POKEA+377-E,C%:POKEA+378-E,B%
140 POKEA+383-E,C%:POKEA+384-E,B%
145 D=A+440:B%=D/256:C%=D-B%*256:POKEA+365-E,C%:POKEA+366-E,B%
150 POKEA+389-E,C%:POKEA+390-E,B%
155 D=A+438:B%=D/256:C%=D-B%*256:POKEA+399-E,C%:POKEA+400-E,B%
160 POKE30845,195:POKE31107,41:POKE31108,122:POKE31104,72
165 POKE31105,122
170 SOUND20,5
195 :
200 REM"PUTGET*3  73 BYTES ****************************************
205 DATA17,0,114
210 DATA183,40,19,254,58,40,15,35,254,49,17,0,116,40,7,254,50
215 DATA17,0,118,32,46,229,33,0,112,24,31,229,33,0,114,17,0,112
220 DATA183,40,21,254,58,40,17,225,35,229,254,49,33,0,116,40
225 DATA7,254,50,33,0,118,32,7,1,0,2,237,176,225,201,225,195
230 DATA151,25
235 :
240 REM"INTERRUPT  20 BYTES ***************************************
245 DATA58,251,104,254,115,40,70,254,121,40,30,254,91,40,93,254
250 DATA122,24,125,0
255 :
260 REM"FILE NAME DATA  21 BYTES *********************************
265 DATA34,83,67,82,69,69
270 DATA78,48,48,34,44,55,48,48,48,44,55,55,70,70,0
275 :
280 REM"IBS  30 BYTES ********************************************
285 DATA33,0,0,205,57,72,33,0,0,126,254,57,48,3,60,119,201,62
290 DATA48,119,43,126,254,57,56,244,62,48,119,201
295 DATA0,0,0,0,0,0
305 :
310 REM"CCS  31 BYTES ********************************************
315 DATA42,32,120,1,0,114,58,24,120,254,0,62,32,32,2,62,96,119
320 DATA50,60,120,35,121,189,32,236,120,188,32,232,201
325 :
330 REM"LDIR:LSTATUS  24 BYTES ***********************************
335 DATA205,196,5,203,71,32,7,62,1,50,156,120,24,15
340 DATA33,230,0,1,75,0,205,92,52,201
345 :
350 REM"DIR:STATUS  12 BYTES *************************************
355 DATA62,0,50,156,120,205,6,73,205,205,82,201
```

```
365 :
370 REM'"END INTERRUPT  13 BYTES ********************************
375 DATA40,242,254,123,192,58,253,104,254,123,40,1,201
395 :
400 REM"DOSWORDS  169 BYTES ***********************************
405 DATA33,3,1,1,75,0,205,92,52,62,13,205,58,3,62,62,205,58,3
410 DATA62,8,205,58,3,205,244,46,254,90,40,44,254,58,48,245,254
415 DATA49,56,241,214,48,245,71,33,229,101,126,254,0,35,32,250
420 DATA16,248,205,80,52,62,8,205,58,3,205,167,40,241,254,9,200
425 DATA33,223,103,24,13,0,62,13,205,58,3,62,13,205,58,3,201
430 DATA121,205,167,40,62,34,205,58,3,6,10,62,8,205,58,3,16,251
435 DATA62,34,205,58,3,201,0,66,83,65,86,69,0,66,76,79,65,68
440 DATA0,66,82,85,78,0,83,65,86,69,0,76,79,65,68,0,82,85,78
445 DATA0,69,82,65,0,68,67,79,80,89,0,83,84,65,84,85,83,0,32
450 DATA32,32,32,32,32,32,32,32,0
455 :
460 REM'"VART  34 BYTES ****************************************
465 DATA35,126,254,54,40,28,254,55,40,21,254,56,40,17,0
470 DATA33,2,114,58,165,120,254,114,32,3,33,233,122,34,164
475 DATA120,195,25,26
495 :
500 REM'"ERT  88 BYTES *****************************************
505 DATA42,164,120,205,1,1,42,249,120,205,1,1,42,251
510 DATA120,205,1,1,42,253,120,205,1,1,42,177,120,205,1,1
515 DATA42,232,120,237,91,253,120,205,1,1,195,25,26,71,82
520 DATA46,32,77,79,68,69,32,50,13,0,0,0,0,0,34,27,120,205,175
525 DATA15,62,32,205,42,3,237,91,164,120,42,27,120,237,82,205
530 DATA175,15,62,13,205,42,3,201
```

SUITE II COMMENTS & INSTRUCTIONS :-

THREE ADDITIONS, DOSWORDS, VART AND ERT, TO THE SUITE OF ROUTINES (SUITE ISSUE 20 OF JOURNAL), MAINLY FOR THOSE OF YOU WHO HAVE DISK DRIVES.

DOSWORDS IS A ROUTINE TO DISPLAY ANY ONE OF NINE VZ DISK WORDS ALONG WITH QUOTES ENCLOSING EIGHT BLANKS FOR TYPING IN A FILE NAME. DOSWORDS IS A SUBROUTINE OF THE INTERRUPT SERVICE ROUTINE OF SUITE2, THAT SAVES YOU THE TROUBLE OF HAVING TO TYPE IN DISK WORDS CHARACTER BY CHARACTER AND THEN HAVING TO USE <SHIFT> <2> TO ENTER OPENING QUOTES FOR THE FILE NAME THEN <SHIFT> <2> AGAIN FOR CLOSING QUOTES.

THE INTERRUPT SERVICE KEYS ASSIGNED TO DOSWORDS ARE :-

<SHIFT> <CTRL>:

HOLD DOWN <SHIFT> AND PRESS <CTRL>, OR SIMPLY PRESS BOTH KEYS TOGETHER WITH THE THUMB.

THE CURSOR WILL MOVE TO START OF A NEW SCREEN LINE AND A > PROMPT WILL APPEAR UNDER THE CURSOR.

YOU CAN NOW PRESS <Z> TO QUIT DOSWORDS IF YOU CHANGE YOUR MIND, OR PRESS A NUMBER KEY <1> TO <9> TO DISPLAY A CORRESPONDING DISK WORD, ALONG WITH OPENING AND CLOSING QUOTES WHERE THESE ARE REQUIRED, AND SUFFICIENT SPACE BETWEEN THE QUOTES TO TYPE IN A FILE NAME. THE CURSOR IS POSITIONED IMMEDIATELY TO THE RIGHT OF THE OPENING QUOTES.

IF THE FILE NAME YOU TYPE IN IS LESS THAN EIGHT CHARACTERS LONG, IT IS NOT NECESSARY TO RUBOUT THE TRAILING SPACES.

<1>	WILL DISPLAY	BSAVE" "
<2>		BLOAD" "
<3>		BRUN" "
<4>		SAVE" "
<5>		LOAD" "
<6>		RUN" "
<7>		ERA" "
<8>		DCOPY" "
<9>		STATUS

THE 2'ND ADDITION TO SUITE IS VART, A SHORT ROUTINE THAT SWITCHES THE START OF BASIC PROGRAM POINTER BETWEEN 31465 AND 29186.

THE SWITCH IS COMMANDED WITH PRINT&.

THE USUAL START ADDRESS OF A BASIC PROGRAM IS 31465, SO A FIRST USE OF PRINT& WILL SWITCH THE POINTER TO 29186, AND A SECOND USE OF PRINT& WILL RESTORE THE POINTER TO 31465.

ADDRESS 29186 IS LOCATED NEAR THE START (29184) OF THE SECOND QUARTER OF VIDEO MEMORY. THE TEXT MODE SCREEN ONLY USES THE FIRST QUARTER OF VIDEO MEMORY (28672 TO 29183), LEAVING FREE THE REMAINING THREE QUARTERS OF VIDEO RAM--1536 CELLS FROM 29184 TO 30719. THIS OTHERWISE WASTED VIDEO MEMORY CAN BE USED TO STORE A BASIC PROGRAM INDEPENDENT OF ANY BASIC PROGRAM IN PROGRAM MEMORY FROM 31465 ONWARD. SO TWO BASIC PROGRAMS CAN RESIDE IN YOUR VZ AT THE SAME TIME AND YOU CAN SWITCH BETWEEN THE TWO WITH THE PRINT& COMMAND OF SUITE2.

PROGRAMS IN VIDEO MEMORY (VPROGS) CAN BE UP TO 1534 BYTES LONG (29186 TO 30719). BECAUSE PRINT& ONLY SWITCHES THE START ADDRESS OF BASIC PROGRAM, THE END OF BASIC PROGRAM REMAINS IN PROGRAM MEMORY AT THE END OF WHATEVER BASIC PROGRAM IS CURRENTLY RESIDENT IN PROGRAM MEMORY.

THIS MEANS THAT WHEN A VPROG IS RUN, ANY VARIABLES USED BY THE PROGRAM ARE SET UP IN VARIABLE LIST TABLES IN PROGRAM MEMORY RATHER THAN AT THE END OF THE VPROG AND SO DO NOT USE UP ANY VIDEO MEMORY OR RISK OVERWRITING AND SO CORRUPTING THE COMMUNICATIONS REGION WHICH COMMENCES AT THE END OF VIDEO MEMORY (30720 TO 31464).

VPROGS ARE BSAVED TO DISK AS BINARY FILES. BINARY FILE FORMAT IS USED SO THAT YOU CAN LOAD VPROGS USING THE BLOAD COMMAND. BLOAD DOES NOT ALTER START AND END OF PROGRAM POINTERS. SO YOU CAN LOAD A MAIN BASIC PROGRAM INTO PROGRAM MEMORY FROM 31465 ONWARD, WITH POINTERS POINTING TO THIS PROGRAM THEN, WHENEVER YOU WISH, BLOAD A VPROG INTO VIDEO MEMORY WITHOUT SWITCHING THE POINTERS FROM THE MAIN PROGRAM.

WHEN YOU ARE READY TO RUN THE VPROG YOU NEED ONLY TYPE IN PRINT& AND PRESS <RETURN> TO SWITCH THE START POINTER TO THE VPROG, THEN ENTER RUN AND PRESS <RETURN>.

WHEN YOU HAVE FINISHED USING THE VPROG, THE PRINT& COMMAND WILL RETURN YOU TO THE MAIN PROGRAM.

CONTINUED IN NEXT ISSUE . . .

FOR SALE E&F W.P.PATCH 3.3

PATCH 3.3 WRITTEN BY DAVE MITCHELL WILL CONVERT YOUR E & F TAPE WORD PROCESSOR FOR FULL DISK USE WHILE RETAINING ALL ORIGINAL FUNCTIONS. BELOW ARE ADDED DISK COMMANDS & FUNCTIONS :-

LOAD, SAVE, ERASE, RENAME, DIRECTORY, INITIALIZE, UPDATE, DRIVE 1 & 2, SHIFTLOCK & IMBEDDED PRINTER CONTROL CODES PLUS CTRL+P WHICH BYPASSES PRINT MENU AND PRINTS TO SCREEN OR PRINTER. A ROUTINE IS ALSO PROVIDED TO CONVERT YOUR BASIC PROGRAM OR SOURCE CODE FILES INTO WORD PROCESSOR FILES.

PATCH 3.3 HAS PROVISION FOR IMBEDDING PRINTER CONTROL CODES IN TEXT AND FAST SAVING AND LOADING OF TEXT DATA TO AND FROM DISK USING BLOCK SAVE/LOAD TECHNIQUES. PRINTER CONTROL CODES CAN BE SAVED TO TAPE OR DISK.

BSTWP.F - THIS UTILITY PROVIDED WITH PATCH 3.3 WILL CONVERT BASIC PROGRAMS AND ED/ASS. SOURCE CODE FILES INTO WORD PROCESSOR FILES.

SYSTEM REQUIREMENTS - VZ 300 + 16K RAM PACK - VZ 200 + 26K

PATCH 3.3 IS COPYRIGHT TO AND ONLY AVAILABLE FROM :-
HUNTER VALLEY VZ USERS' GROUP P.O.BOX 161 JESMOND 2299
N.S.W. AUSTRALIA - PHONE JOE LEON (049) 51 2756

PRICE - AUS/NZ AU$20.00 - UPDATE - AUS-$10.00 - NZ-AU$11.00.
UPDATING AVAILABLE ONLY TO PREVIOUS PURCHASERS OF PATCHES.

FOR MORE INFORMATION WRITE TO H.V.VZ.U.G. ENCLOSING A SSAE.

EXTENDED DOS V1.3 - $15.00

UPDATED VERSION WITH EXTRA COMMANDS ADDED :-

OLD COMMANDS - MERGE, DIRA, LDIRA, DIRB, LDIRB, OLD, OLD., DEC, HEX, STATUSA AND LSTATUSA. STATUSA AND LSTATUSA ALSO WORKS WITH VERSION 1.0 DOS.

NEW COMMANDS :-

MENU - LOADS AND RUNS BINARY OR TEXT MENU PROGRAM FROM DISK.
CODE - SIMPLIFIES USING PRINTER CONTROL CODES DIRECTLY OR FROM WITHIN A PROGRAM.
LTAB - IS FOR SETTING OF LEFT MARGIN.
MOVE - MOVES BASIC FILE FROM DISK TO CHOSEN MEMORY ADDRESS.
UPD - ERASES OLD FILE AND SAVES WITH SAME FILE NAME.

MENU/FILE COPIER - $15.00

THIS UTILITY WILL READ YOUR DISK DIRECTORY AND PRESENT YOU WITH SEVERAL OPTIONS. USING THE CURSOR YOU CAN RUN/BRUN ANY PROGRAM OR SELECT FILE COPY, REN, ERASE, DRIVE 1 OR 2, ETC. BESIDES COPYING TEXT AND BINARY FILES ALL OTHER FILES CAN BE COPIED AS WELL EXEPT FOR DATA FILES.

FOR PURCHASE OR INFORMATION CONTACT DAVE MITCHELL - (079) 27 8519
24 ELPHINSTONE ST. NORTH ROCKHAMPTON QUEENSLAND 4701

FOR INFORMATION OR DEMONSTRATION IN NEWCASTLE AREA CONTACT :-
JOE LEON - (049) 51 2756 - 22 DRURY ST. WALLSEND NSW 2287

CONTRIBUTIONS TO THE HUNTER VALLEY VZ JOURNAL :-

IF YOU ARE THINKING OF CONTRIBUTING TO THE JOURNAL THE PREFERED FORMAT IS BASIC LISTINGS, WORD PROCESSOR OR SOURCE CODE FILES ON TAPE OR DISK. FILES FROM THE FOLLOWING WORD PROCESSORS CAN BE ACCEPTED :-

E & F TAPE OR DISK PATCH 3.1-3.3, WORDPRO CARTRIDGE, WORDPRO PATCH AND ALL QUICKWRITE WORD PROCESSOR FILES.

WANTED TO BUY ------------------

64K RAM PACKS & VZ200 6K RAM BOARDS - CONTACT JOE LEON
22 DRURY STREET WALLSEND NSW 2287 --- PHONE (049) 51 2756

CLUB MEETINGS -- ALL WELCOME --

FIRST FRIDAY OF MONTH - NO MEETING IN JANUARY 1990

VENUE - JESMOND NEIGHBOURHOOD CENTRE MORDUE PARADE JESMOND
(REAR STOCKLAND MALL - BIG W)

OCTOBER 6 - SPEECH SYNTHESISER
NOVEMBER 3 - EPROM PROGRAMMER & ERASER
DECEMBER 1 - CHRISTMAS MEETING - BRING FAMILY, FRIENDS & PLATE
DECEMBER ~~8~~ 10 - FAMILY PICNIC AT SPEERS POINT PARK - BYO FOOD/GEAR

FUTURE DEMONSTRATIONS - AUCTION NIGHT - USING THE VZ, RITTY, ETC.
IF YOU HAVE ANY IDEAS FOR A DEMONSTRATION OR A SUBJECT THEN PLEASE LET YOUR COMMITTEE KNOW.

CLUB COMMITTEE & SUBSCRIPTIONS -

PRESIDENT --------- ROSS WOODS --- (049) 71 2843
SECRETARY/EDITOR -- JOE LEON ----- (049) 51 2756
TREASURER --------- GARY BULLEY -- (049) 54 7561
COMMITTEE MEMBERS - COLIN BRIDGE - ANDREW IRVINE - PETER JONES

SUBSCRIPTION TO - AUST. - 6 MONTHS $11.00 - 12 MONTHS $21.00
H.V.VZ.JOURNAL - N. Z. - 6 MONTHS $13.00 - 12 MONTHS $26.00

HUNTER VALLEY VZ USERS' GROUP - P.O. BOX 161 JESMOND 2299
NEW SOUTH WALES AUSTRALIA

VZ USER GROUPS & PUBLICATIONS --

J.C.E. D'ALTON 39 AGNES ST. TOOWONG QUEENSLAND 4066
LE'VZ OOP (VZ MAGAZINE) - VSOFTWAREZ/SOFTWARE/HARDWARE FOR SALE

VZ DOWN UNDER - VZ MAGAZINE - 6 ISSUES - $18.00 PER YEAR
HARRY HUGGINS 12 THOMAS ST. MITCHAM 3132 VICTORIA

WAVZ - WESTERN AUSTRALIA VZ USER GROUP
GRAEME BYWATER P.O. BOX 388, MORLEY W.A. 6062

BRISBANE VZ USERS WORKSHOP - C/O 63 TINGALPA ST. WYNUM WEST 4178
SOFTWARE FOR SALE - DISK MENU

NOTE :- WHEN WRITING TO ANY ABOVE OR H.V.VZ. USERS' GROUP FOR INFORMATION PLEASE ENCLOSE A S.S.A.E. OR NZ 2 INT. REPLY COUPONS.

IPL SEQUENCE DECODED BY R. QUINN

FORMAT OF DISASSEMBLY -

EACH MACHINE CODE INSTRUCTION IS DISASSEMBLED AND PRINTED OUT ON TWO OR MORE LINES. THE FIRST LINE DISPLAYS THE START ADDRESS OF THE INSTRUCTION IN DECIMAL THEN HEX THEN, ON THE SAME LINE, THE CONTENT BYTE OF THAT ADDRESS IN HEX THEN DECIMAL.

THE OPCODE AND OPERAND THAT ARE THE ASSEMBLER LANGUAGE INTERPRETATION OF THE INSTRUCTION WILL APPEAR ON THE NEXT LINE. DIGITS OR LETTER CHARACTERS A TO F THAT APPEAR IN THE OPERAND PRECEDED BY A HASH SIGN (#) ARE HEX NUMBERS.

LETTER CHARACTERS IN THE OPERAND THAT ARE NOT PRECEDED BY A HASH SIGN ARE REGISTERS OR, IN THE CASE OF JUMP INSTRUCTIONS, FLAG CONDITIONS.

NUMBERS IN THE OPERAND THAT ARE PRECEDED BY A DOLLAR SIGN ($) ARE DECIMAL NUMBERS.

THE BYTE OR BYTES OF THE OPERAND, IF SUCH THERE BE, WILL DISPLAY ON SUCCESSIVE LINES WITH THEIR DECIMAL ADDRESSES. COMPLETION OF THE DISASSEMBLY OF AN INSTRUCTION IS MARKED BY A LINE OF DASHES.

USING INFO FROM IPL SEQUENCE —

ON PAGE 26/21 ON IPL SEQUENCE DECODING IN THIS ISSUE YOU'LL NOTE ADDRESSES 181 TO 189 INCLUSIVE HAVE 9 NOP INSTRUCTIONS WHICH CAN BE UTILISED. WITH ROBERT'S HELP I'VE PLACED FOUR BYTES THERE WHICH PERFORM AN OUT 32,8 FUNCTION UPON POWER UP OR RESET. AS ONLY FOUR BYTES WERE USED THAT MEANS THERE ARE FIVE BYTES LEFT THAT COULD BE USED TO ADD OTHER FUNCTIONS.

DECIMAL ADDRESS	OLD BYTE	NEW BYTE		
181	0	62	;	LD A
182	0	8	;	8
183	0	211	;	OUT
184	0	32	;	(32),A
185	0	0		
186	0	0		
187	0	0		
188	0	0		
189	0	0		

A VZ MODIFIED FOR SUPER HI-RES GRAPHICS POWERS UP OR RESETS IN GRAPHICS MODE(0) WITH ONLY 1K (1024) BYTES VIDEO MEMORY AVAILABLE. THE USER HAS TO DO AN OUT 32.8 FUNCTION TO PUT THE VZ INTO GRAPHICS MODE(2) WHICH IS THE STANDARD MODE GIVING 2K (2048) BYTES FOR HI-RES GRAPHICS. SO HAVING THE VZ DO AN OUT 32,8 AUTOMATICALLY FOR USER IS A BIG ADVANTAGE.

NOTE - AS THE ADDRESSES 181 TO 189 ARE IN ROM POKEING WILL NOT CHANGE ANY BYTE AND ONLY THOSE PERSONS WITH AN EPROM PROGRAMMER OR BATTERY BACKED RAM WILL BE ABLE TO MODIFY THEIR ROMS.

IPL SEQUENCE DECODING CONTINUED . . 26/20

THE INITIAL PROGRAM LOADER (IPL) SEQUENCE EXECUTES WHEN VZ IS SWITCHED ON

```
________________________________
 0     00       F3      243   INTERRUPTS DISABLED
DI
--------------------------------
 1     01       AF      175   ZERO THE ACCUMULATOR
XOR            A
--------------------------------
 2     02       32       50   RAM LOCATION
LD             (#6800),A
 3              00        0   #6800 = 26624:
 4              68      104   IS ZEROED
--------------------------------
 5     05       C3      195   #0674 = 1652:
JP             #0674
 6              74      116   JUMP TO CONTINUE THE
 7              06        6   IPL SEQUENCE
================================
 1652  674      00        0
NOP
--------------------------------
 1653  675      00        0
NOP
--------------------------------
 1654  676      21       33   SETTING UP FOR A COPY
LD             HL,#06D2
 1655           D2      210   FROM ROM STARTING AT
 1656           06        6   #06D2 = 1746,
--------------------------------
 1657  679      11       17   TO RAM
LD             DE,#7800
 1658           00        0   #7800 = 30720: IS START
 1659           78      120   OF COMMUNICATIONS REGION
--------------------------------
 1660  67C      01        1   OF 54 BYTES
LD             BC,#0036
 1661           36       54   #6D2 TO #707 WILL COPY
 1662           00        0   INTO #7800 TO 7835
--------------------------------
 1663  67F      ED      237   SEE COMMUNICATIONS REGION
LDIR
 1664           B0      176   SUPPLEMENT IN ISSUE 20
--------------------------------
 1665  681      3D       61   A = 255 WHEN DECREMENTED
DEC            A
--------------------------------
 1666  682      3D       61   FROM ZERO:DECREMENT TWICE
DEC            A
--------------------------------
 1667  683      20       32   JUMP TO 1654 AND REPEAT
JR             NZ,$ -15
 1668           F1      241   COPY IF REGISTER A NOT 0
================================
 1669  685      06        6   COPY REPEATED 128 TIMES!
LD             B,#27
 1670           27       39   39 LOCATIONS FROM #7836
--------------------------------
 1671  687      12       18   WILL BE ZEROED
LD             (DE),A
--------------------------------
```

```
 1672  688      13       19   IN COMMUNICATIONS REGION
INC            DE
--------------------------------
 1673  689      10       16   JUMP TO 1671
DJNZ           $  -4
 1674           FC      252   ZERO #7836 TO #785C
================================
 1675  68B      C3      195   SEE COM.REGION SUPPLEMENT
JP             #0075
 1676           75      117   #0075 = 117: JUMP AND
 1677           00        0   CONTINUE IPL SEQUENCE
================================
 117   75       11       17   #7880 = 30848: START LOC.
LD             DE,#7880
 118            80      128   IN COMMUNICATIONS REGION
 119            78      120   FOR ANOTHER COPY
--------------------------------
 120   78       21       33   FROM ROM, BEGINNING AT
LD             HL,#18F7
 121            F7      247   #18F7 = 6391
 122            18       24
--------------------------------
 123   7B       01        1   OF 39 BYTES: WILL COPY
LD             BC,#0027
 124            27       39   INITIALISATION DATA IN
 125            00        0   ROM #18F7 TO #191D
--------------------------------
 126   7E       ED      237   TO COMMUNICATIONS REGION:
LDIR
 127            B0      176   #7880 TO #78A6
--------------------------------
 128   80       21       33   CONTINUE INITIALISATION
LD             HL,#79E5
 129            E5      229   OF COMMUNICATIONS REGION
 130            79      121   #79E5 = 31205
--------------------------------
 131   83       36       54   PUT ASCII CODE FOR COLON
LD             (HL),#3A
 132            3A       58   IN LOCATION 31205
--------------------------------
 133   85       23       35   HL = 31206
INC            HL
--------------------------------
 134   86       70      112   PUT ZERO IN 31206
LD             (HL),B
--------------------------------
 135   87       23       35
INC            HL
--------------------------------
 136   88       36       54   PUT ASCII CODE FOR COMMA
LD             (HL),#2C
 137            2C       44   IN LOCATION 31207
--------------------------------
 138   8A       23       35   HL = 31208
INC            HL
--------------------------------
```

```
------------------------------------
139   8B      22      34   #78A7 = 30887: KEYBOARD
LD            (#78A7),HL
140           A7     167   BUFFER POINTER SET TO
141           78     120   START OF INPUT BUFFER
------------------------------------
142   8E      11      17   #012D = 301: IS ADDRESS
LD            DE,#012D
143           2D      45   FIELD FOR JP #12D
144           01       1   28 JUMPS WILL BE SET UP
------------------------------------
145   91      06       6   IN COMMUNICATIONS REGION
LD            B,#1C
146           1C      28   FROM 31058 TO 31141:JUMPS
------------------------------------
147   93      21      33   TO ERROR MESSAGE IF TRS80
LD            HL,#7952
148           52      82   DISK COMMAND TOKENS USED
149           79     121   #7952 = 31058: IS START
------------------------------------
150   96      36      54   OF DISK BASIC EXITS IN
LD            (HL),#C3
151           C3     195   THE COMMUNICATIONS REGION
------------------------------------
152   98      23      35   #C3 = 195 IS MACHINE CODE
INC           HL
------------------------------------
153   99      73     115   FOR UNCONDITIONAL JUMP
LD            (HL),E
------------------------------------
154   9A      23      35
INC           HL
------------------------------------
155   9B      72     114
LD            (HL),D
------------------------------------
156   9C      23      35
INC           HL
------------------------------------
157   9D      10      16   JUMP TO 150
DJNZ          $ -9
158           F7     247   LOOP 28 TIMES
====================================
159   9F      06       6   21: LOOP COUNT FOR
LD            B,#15
160           15      21   TRS-80 DOS EXIT RETURNS
------------------------------------
161   A1      36      54   STARTING AT 31142,
LD            (HL),#C9
162           C9     201   INITIALISE 21 RETURNS
------------------------------------
163   A3      23      35   THE FIRST TWO RETURNS ARE
INC           HL
------------------------------------
164   A4      23      35   FOR DISK BASIC EXITS
INC           HL
------------------------------------
```

```
------------------------------------
165   A5      23      35   DOS EXIT RETURNS BEGIN AT
INC           HL
------------------------------------
166   A6      10      16   31148
DJNZ          $ -7
167           F9     249   JUMP TO 161:LOOP 21 TIMES
====================================
168   A8      21      33   #7AE8 = 31464: IS LAST
LD            HL,#7AE8
169           E8     232   LOCATION IN COM. REGION
170           7A     122   ZERO THIS LOCATION
------------------------------------
171   AB      70     112   BECAUSE BASIC PROGRAMS
LD            (HL),B
------------------------------------
172   AC      31      49   REQUIRE 0 START TO RUN
LD            SP,#79F8
173           F8     248   #79F8 = 31224: IS
174           79     121   TEMPORARY START OF STACK
------------------------------------
175   AF      CD     205   DURING IPL SEQUENCE
CALL          #1B8F
176           8F     143   #1B8F = 7055: INITIALISE
177           1B      27   BASIC POINTERS/VARIABLES
====================================
178   B2      CD     205
CALL          #01C9
179           C9     201   #01C9 = 457: TO CLEAR
180           01       1              THE SCREEN
====================================
181   B5      00       0   THESE NOP INSTRUCTIONS
NOP
------------------------------------
182   B6      00       0   WERE THE LOCATION OF A
NOP
------------------------------------
183   B7      00       0   TRS-80 USER INPUT MEMORY
NOP
------------------------------------
184   B8      00       0   SIZE ROUTINE THAT HAS
NOP
------------------------------------
185   B9      00       0   BEEN OBLITERATED FROM THE
NOP
------------------------------------
186   BA      00       0   VZ OPERATING SYSTEM
NOP
------------------------------------
187   BB      00       0
NOP
------------------------------------
188   BC      00       0
NOP
------------------------------------
189   BD      00       0
NOP
------------------------------------
```

IPL SEQUENCE DECODING CONTINUED . . 26/22

```
-------------------------------
190   BE       18        24   JUMP TO 196
JR             $ +4
191            04         4
===============================
192   C0       D7       215   NOT PART OF VZ IPL
RST            #10
===============================
193   C1       B7       183   NOT PART OF VZ IPL
OR             A
-------------------------------
194   C2       20        32
JR             NZ,$ +18
195            12        18   NOT PART OF VZ IPL
===============================
196   C4       21        33   #7B4C = 31564: SETTING UP
LD             HL,#7B4C
197            4C        76   FOR A ROUTINE TO
198            7B       123   DETERMINE MEMORY SIZE
-------------------------------
199   C7       23        35   STARTING AT 31565 IN RAM
INC            HL
-------------------------------
200   C8       7C       124   AND WORKING TOWARD 65535
LD             A,H
-------------------------------
201   C9       B5       181   (MAXIMUM ADDRESSABLE
OR             L
-------------------------------
202   CA       28        40   MEMORY FOR VZ), TESTING
JR             Z,$ +27
203            1B        27   FOR PRESENCE OF RAM
===============================
204   CC       7E       126   GET BYTE AT CURRENT LOC.
LD             A,(HL)
-------------------------------
205   CD       47        71   BEING TESTED AND SAVE IT
LD             B,A
-------------------------------
206   CE       2F        47   COMPLEMENT IT & WRITE NEW
CPL
-------------------------------
207   CF       77       119   BYTE TO TEST LOCATION
LD             (HL),A
-------------------------------
208   D0       BE       190   CHECK IF WRITE SUCCEEDED
CP             (HL)
-------------------------------
209   D1       70       112   RESTORE ORIGINAL BYTE
LD             (HL),B
-------------------------------
210   D2       28        40   IF TEST LOCATION EXISTS,
JR             Z,$ -13
211            F3       243   JUMP TO 199 & TEST NEXT
===============================
```

```
-------------------------------
212   D4       18        24   JUMP TO 231 BECAUSE
JR             $ +17
213            11        17   CURRENT TEST LOC. ABSENT
===============================
===============================
231   E7       2B        43   WE HAVE TOP OF RAM
DEC            HL
-------------------------------
232   E8       11        17   #7C14 = 31764: USE THIS
LD             DE,#7C14
233            14        20   AS MINIMUM REQUIRED RAM
234            7C       124   AND GO TEST THAT MINIMUM
-------------------------------
235   EB       DF       223   RAM NOT LESS THAN 31764
RST            #18
===============================
236   EC       DA       218   #197A = 6522: JUMP TO
JP             C,#197A
237            7A       122   INSUFFICIENT MEMORY ROUT-
238            19        25   INE IF MINIMUM RAM LESS
===============================
239   EF       11        17   ELSE CONTINUE IPL
LD             DE,#FFCE
240            CE       206   #FFCE = 65486:-
241            FF       255   DEFAULT SIZE (50) FOR
-------------------------------
242   F2       22        34   STRING SPACE (65536-50)
LD             (#78B1),HL
243            B1       177   #78B1 = 30897: SET
244            78       120   TOP OF RAM POINTER
-------------------------------
245   F5       19        25
ADD            HL,DE
-------------------------------
246   F6       22        34   #78A0 = 30880: SET LOWER
LD             (#78A0),HL
247            A0       160   BOUNDARY OF STRING SPACE
248            78       120   IN POINTER 30880/1
-------------------------------
249   F9       CD       205   #1B4D = 6989: GO AND
CALL           #1B4D
250            4D        77   INITIALISE ALL BASIC
251            1B        27   POINTERS AND VARIABLES
===============================
252   FC       CD       205   #3484 = 13444:
CALL           #3484
253            84       132   SEE DISASSEMBLY AT END
254            34        52   OF THIS PRINTOUT
===============================
255   FF       21        33   #010F = 271: IS START OF
LD             HL,#010F
256            0F        15   VIDEO TECHNOLOGY MESSAGE
257            01         1
-------------------------------
258   102      CD       205   #28A7 = 10407: TO MESSAGE
CALL           #28A7
259            A7       167   OUTPUT ROUTINE
260            28        40
===============================
```

IPL SEQUENCE DECODING CONTINUED . . 26/23

```
 261   105   ED        237
IM          1
 262          56        86       SET INTERRUPT MODE 1 CONDITION--A RST #38 EXECUTES IF AN INTERRUPT OCCURS
-------------------------------
 263   107   C3        195
JP         #068E
 264          8E       142       CONTINUE IPL SEQUENCE FROM #68E
 265          06         6
-------------------------------
-------------------------------
 1678  68E   21         33
LD         HL,#4000
 1679         00          0      #4000 = 16384  IS START OF DOS ROM IF DISK DRIVE INSTALLED
 1680         40         64
-------------------------------

 1681  691   CD        205       GO AND CHECK FOR PRESENCE OF DOS BY TESTING IDENTITY OF FIRST FOUR BYTES FROM 16384
CALL       #06A4
 1682         A4        164      IF TEST REVEALS PRESENCE OF DOS, THE CHECKING SUBROUTINE WILL NOT RETURN
 1683         06          6      INSTEAD, A JUMP TO DOS IPL SEQUENCE WILL BE EXECUTED
-------------------------------
 1684  694   21         33       NO DOS, SO RETURN HERE
LD         HL,#6000
 1685         00          0      #6000 = 24576 IS THE START OF THE 2K LOCATED ABOVE DOS ROM
 1686         60         96
-------------------------------
 1687  697   CD        205       GO AND CHECK FOR PRESENCE OF FOUR IDENTITY BYTES FROM 24576
CALL       #06A4
 1688         A4        164      IF ALL BYTES FOUND THEN CHECKING ROUTINE WILL NOT RETURN
 1689         06          6      INSTEAD A JUMP TO 24580 WILL BE EXECUTED
-------------------------------
 1690  69A   21         33       #8000 = 32768  IS USUALLY AN ADDRESS IN PROGRAM (RAM) MEMORY
LD         HL,#8000
 1691         00          0
 1692         80        128
-------------------------------
 1693  69D   CD        205       GO AND CHECK FOR PRESENCE OF FOUR IDENTITY BYTES FROM 32768
CALL       #06A4
 1694         A4        164      UNLESS ROM OR EPROM HAS BEEN INSTALLED FROM 32768 WITH THE FOUR IDENTITY BYTES,
 1695         06          6      THE TEST WILL FAIL AND CHECKING SUBROUTINE WILL RETURN TO CALLER
-------------------------------
 1696  6A0   FB        251       INTERRUPTS WILL BE ENABLED
EI
-------------------------------
 1697  6A1   C3        195       AND THE IPL SEQUENCE WILL END WITH A JUMP TO READY ROUTINE
JP         #1A19
 1698         19         25      YOUR VZ IS SET UP FOR YOUR USE
 1699         1A         26
-------------------------------
-------------------------------
```

CONTINUED NEXT ISSUE . . . .